

# NFT Floor Price Prediction

Laurea Magistrale in Informatica 8028

CORSO DI COMPLEMENTI DI BASI DI DATI -  
INTELLIGENZA ARTIFICIALE (C.I.) a.a. 2021/2022



## Gruppo Creative

[alex.citeroni@studio.unibo.it](mailto:alex.citeroni@studio.unibo.it)

Branchetti Simone [1038801]

Citeroni Alex [1052175]

Talevi Davide [1043409]

Scozia Mirko [1053220]

## Indice

<b>Introduzione</b>	<b>3</b>
Descrizione del problema	3
Descrizione della soluzione proposta	3
La rivoluzione degli NFT	3
Cos'è un NFT?	4
Cos'è una collezione di NFT?	5
Dove e come vengono scambiati gli NFT?	6
Perché il Floor Price è l'unico dato che utilizziamo?	6
Revisione della letteratura	7
Collaborazione	7
Presentazione dei risultati ottenuti	7
Preparazione dei dati	8
<b>Metodi proposti</b>	<b>10</b>
Alberi di regressione	11
Modello AutoRegressivo	11
Modello ARMA	12
Modello ARIMA	13
<b>Risultati sperimentali</b>	<b>15</b>
Risultati della configurazione migliore	18
Studio di ablazione	19
Studio di comparazione	20
<b>Conclusione</b>	<b>21</b>
Limiti del metodo	21
Lavori futuri	22
<b>Sitografia</b>	<b>23</b>

## Introduzione

### Descrizione del problema

Il problema principale che cerchiamo di risolvere con questo progetto è quello di trovare (se esiste) un modo per **prevedere** il Floor Price (prezzo minimo a cui comprare un NFT in una collezione) futuro di una collezione di NFT.

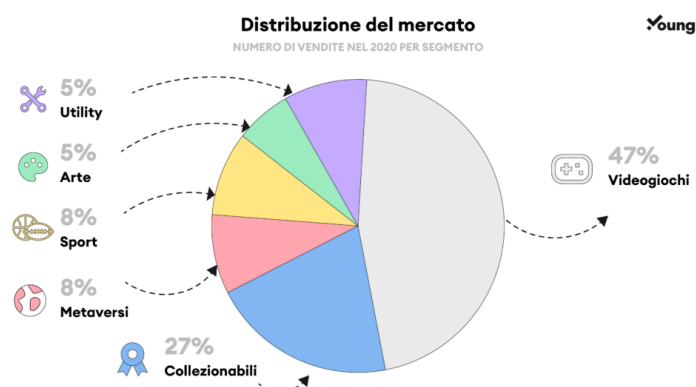
Il mondo degli NFT è relativamente **nuovo** e notoriamente difficile da navigare per l'utente medio, tanto che molti neofiti vedono lo scambio di questi token come una truffa, soprattutto a causa della presunta volatilità del mercato, spesso attribuita a fattori incontrollabili. Questo progetto mira a fornire un **tool**, per quanto semplice, che possa aiutare l'utente medio e/o i trader esperti a predire l'andamento futuro del Floor Price di una collezione di NFT.

### Descrizione della soluzione proposta

Per raggiungere l'obiettivo che ci siamo posti utilizzeremo un programma di Intelligenza Artificiale scritto in Python basato sul modello di regressione lineare, chiamato ARIMA, e utilizzeremo (tramite Covalent) dati provenienti dai più grandi marketplace di NFT esistenti al momento, come **OpenSea**, dove vengono scambiati **milioni di dollari** in NFT ogni giorno.

### La rivoluzione degli NFT

Gli NFT sono la base di una vera e propria **rivoluzione** nel mondo digitale, che non si limita solo al mercato dell'arte. L'adozione degli NFT ha già raggiunto il mondo dei videogiochi e diversi altri settori coinvolti nella produzione di contenuti digitali come musica e video. Complessivamente, nei primi quattro mesi del 2021, il volume degli NFT ha superato i **2 miliardi di dollari**, dieci volte superiore all'intero volume degli scambi nell'intero 2020.



Un esempio di NFT utilizzato per rappresentare l'arte digitale è **CryptoKitties**, un gioco sulla blockchain di **Ethereum** che consente agli utenti di acquistare, collezionare, allevare e vendere gatti virtuali. Nel dicembre del 2017, erano presenti talmente tanti utenti che il gioco è riuscito a **congestionare** la rete Ethereum.

Nel marzo 2021, l'artista noto come Beeple ha venduto un NFT del suo ultimo lavoro per **69,3 milioni di dollari**. L'acquisto ha portato al terzo prezzo d'asta più alto raggiunto da un artista vivente, dopo Jeff Koons e David Hockney.

## Cos'è un NFT?

Un NFT, acronimo di “**Non Fungible Token**”, in italiano “Token Non Fungibile” è unità di dati univoca che può essere utilizzata per certificare la **proprietà** di una risorsa digitale.

Gli NFT permettono di stabilire la **provenienza** di un oggetto digitale, offrendo risposte indiscutibili su chi lo possiede attualmente, chi l'ha posseduto precedentemente e su chi lo ha creato, nonché quale delle molte copie è l'originale. Questi vengono creati su **blockchain** utilizzando smart contract (solitamente viene utilizzata la blockchain di **Ethereum**) e possono rappresentare **qualsiasi oggetto** digitale, incluse foto, gif, opere d'arte, meme, musica, oggetti di gioco e da collezione, persino tweet.

La novità degli NFT rispetto agli oggetti d'arte “fisici” è che gli asset digitali possono in teoria essere replicati e scaricati un numero illimitato di volte, e possono quindi essere “posseduti” da chiunque. **Possedere** un NFT infatti, significa che in un blocco di una particolare blockchain è registrata la transazione di vendita, con il proprio portafoglio all'indirizzo del compratore.

La rarità è una caratteristica insita agli NFT che per natura sono oggetti unici. Alcuni NFT però sono **più unici di altri**, e questo è ciò che intendiamo quando parliamo di rarità.

Se prendiamo ad esempio una collezione come quella di **Cryptopunks** che ha 10.000 personaggi totalmente unici, alcuni avranno dei tratti e delle caratteristiche che li renderanno più rari di altri della stessa collezione.



## Cos'è una collezione di NFT?

Una collezione di NFT è un **gruppo** di NFT, creato dallo stesso autore sotto lo stesso nome e rappresenta un modo facile ed immediato per raggruppare NFT in tutti i mercati virtuali.

Una delle più famose collezioni è **Bored Ape Yacht Club**: una collezione di **10.000 token non fungibili** rilasciata sul mercato da **Yuga Labs** nel 2021. Ogni NFT della collezione rappresenta un **"Ape"**, ovvero una scimmia, **generata dalla combinazione di 170 caratteristiche scelte randomicamente**, tra le quali ci sono le espressioni, i copricapi e l'abbigliamento.

Siccome ogni collezione è composta da NFT che vengono venduti e comprati regolarmente, abbiamo bisogno di alcune misure del successo e del market cap di ognuna. Una di queste misure è il **Floor Price**: il prezzo minore richiesto per possedere un NFT appartenente a una data collezione. Per Bored Ape, al 03/09/2022 il Floor Price è di **77.77 ETH** (121.638,44 €).



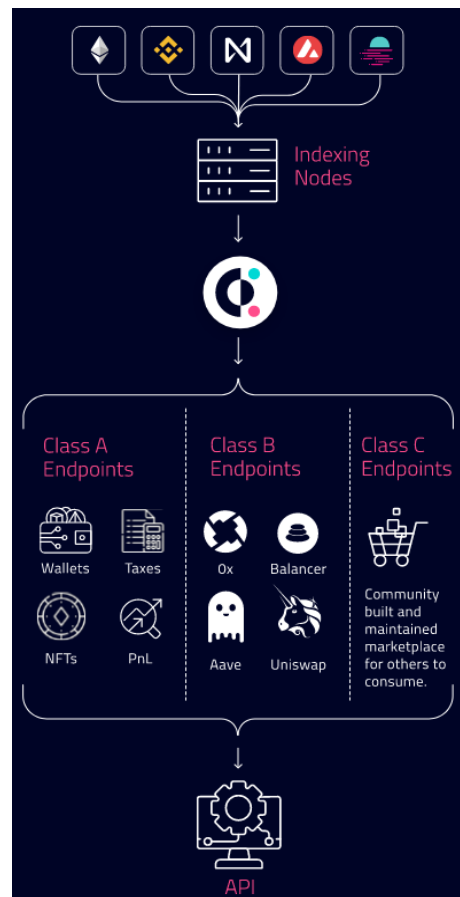
## Dove e come vengono scambiati gli NFT?

Gli NFT possono essere acquistati tramite **criptovalute** che cambiano in base alla blockchain su cui ci troviamo; la più utilizzata per gli scambi di NFT è la blockchain di Ethereum e in questo caso viene utilizzata principalmente la criptovaluta **ETH**.

Esistono molti mercati virtuali su cui è possibile scambiare NFT, il più famoso tra i quali è **OpenSea**. Ogni mercato permette scambi su una o più blockchain (**Ethereum, Solana, Polygon**, etc).

**OpenSea**, come tanti altri, è un mercato incentrato sulle transazioni e consente l'accesso alle API di terze parti per il recupero dei dati.

Per accedere ai dati dei maggiori mercati di NFT utilizziamo l'API **Covalent**, che estrae i dati da più di 40 blockchain e ci permette di visualizzare ad esempio il Market Cap, il volume di vendite in 24h in USD, il market Cap Domination, il Chain ID, il Collection Address e soprattutto il Floor Price di molte collezioni di NFT presenti su OpenSea, Rarible, Refinable, NFTtrade, ecc...



## Perchè il Floor Price è l'unico dato che utilizziamo?

Nel mercato degli NFT, il floor price è la **quantità minima** di denaro che puoi spendere per diventare membro di un progetto (possedere un NFT). Il Floor Price è stabilito dall'individuo che possiede un NFT in una collezione specifica e lo mette in vendita al prezzo più basso di tutte le altre persone all'interno di quella data collezione.

**Floor Price** è un termine che è stato comunemente usato dagli economisti per rappresentare quanto basso può essere il **prezzo** per una certa merce, bene o servizio. Nel mercato NFT, è una delle **metriche più usate** per valutare un progetto. L'acquisto di un NFT al prezzo minimo è spesso fatto per ricevere i **benefici** di essere un proprietario (come l'accesso prioritario ai rilasci futuri dello stesso autore). Gli NFT all'interno di un progetto che vengono venduti al prezzo minimo o vicino al prezzo minimo spesso hanno proprietà o caratteristiche che **non sono rare** e quindi non creano valore aggiunto però sono comunque un indicatore importante dell'andamento del prezzo dell'intera collezione.

Abbiamo scelto di considerare **solamente** il Floor Price di una collezione di NFT per semplificare il problema a un dato comune a tutte le collezioni che rappresenta comunque un'informazione interessante e facilmente comprensibile.

## Revisione della letteratura

- [Wang et al.](#) e [Nadini et al.](#) , nel 2021, ci danno un'infarinatura sul mondo degli NFT, elencando le **tecnologie** in uso, le sfide ancora da affrontare e un overview sulla sicurezza di questo mercato.
- [Jain et al.](#) , nel loro articolo del 2022 esplorano la **relazione** tra NFT e varie caratteristiche divise in 3 categorie: dati del mercato, metadata degli NFT e trend dei social media.
- [Durban Rodrigues](#), nel 2022 usa il modello LSTM per creare un bot che **prevede** l'andamento del prezzo medio di diverse collezioni di NFT. Questo approccio usa una rete neurale e si concentra su alcune tra le più famose collezioni di NFT al mondo, con dati reperiti da OpenSea.
- [Ho, Kin-Hon, et al.](#), usando l'algoritmo XGBoost provano che l'**utilità** di un NFT legato a un gioco play-to-earn (Axie Infinity) è un fattore importante nella previsione del prezzo del token.

Il nostro lavoro si posiziona a fianco degli ultimi proposti, come **previsione** di una variabile ancora non considerata a fondo quale il Floor Price e come **esplorazione** di modelli lineari e la loro validità in questo caso di studio.

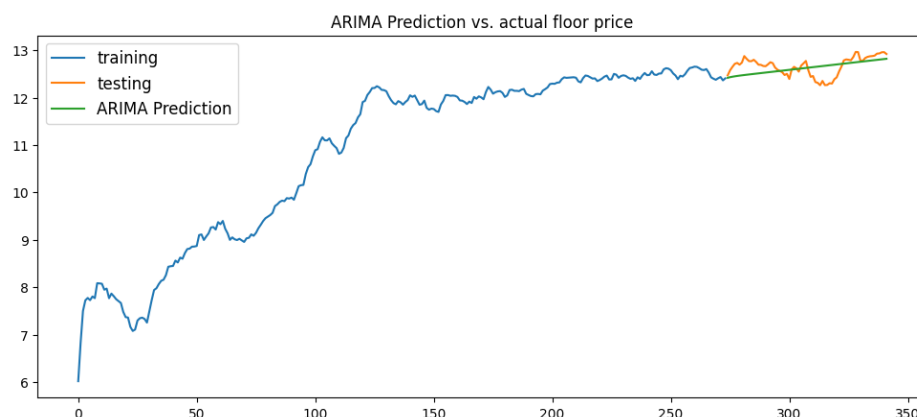
## Collaborazione

Come prima fase di sviluppo ci siamo concentrati ad avere una **visione chiara** del problema e delle entità in campo. Sono stati poi suddivisi tra i componenti del gruppo i modelli analizzati in questo studio per stilare valutazioni al riguardo e capire quali fossero adatti e quali meno. Dei modelli che abbiamo selezionato per il nostro studio Davide e Mirko hanno analizzato principalmente il modello AR mentre Simone e Alex si sono occupati del modello ARMA e ARIMA. Abbiamo poi unito i risultati degli studi facendo una valutazione su quale modello performa meglio per il nostro problema.

## Presentazione dei risultati ottenuti

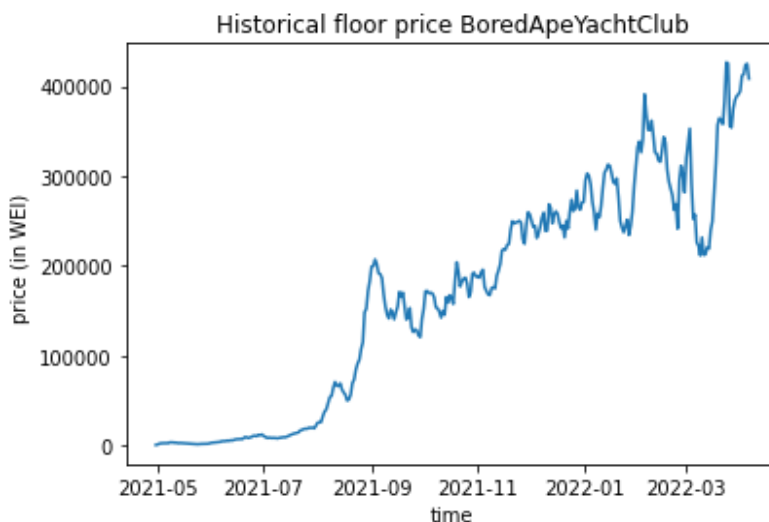
Con questo studio si è rilevata la possibilità di avere una **previsione** certa al **98,76%** del prezzo futuro di una collezione di NFT, grazie all'utilizzo del modello che a noi è sembrato il migliore per il problema in questione, cioè il modello ARIMA(2, 2, 1).

In questo grafico si mostra la previsione del floor price per i dati di test partizionati dal nostro dataset.

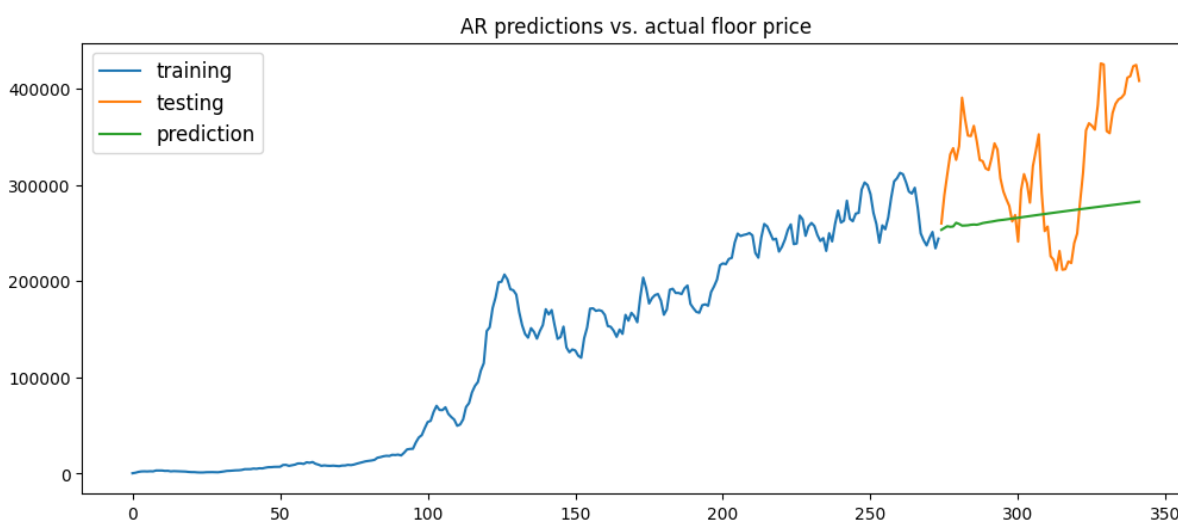


## Preparazione dei dati

Dall'immagine seguente, possiamo osservare il grafico dello storico del Floor Price della collezione NFT di Bored Ape Yacht Club in formato **WEI** (unità di misura di ETH), cioè con i dati in formato standard forniti dall'API di Covalent.



Possiamo notare la grande crescita dei prezzi nel tempo e soprattutto la **grande varianza** che li caratterizza. Questi dati, se inseriti in un modello qualunque di previsione, possono portare a risultati meno precisi, come possiamo notare dal seguente grafico creato in fase di sperimentazione attraverso il modello AR.



Al fine di poter effettuare un'analisi più accurata e precisa, avente una varianza ottimale, siamo passati al **logaritmo naturale** ( $\log_e n$ ) dei prezzi rendendo la serie dei prezzi stazionaria.

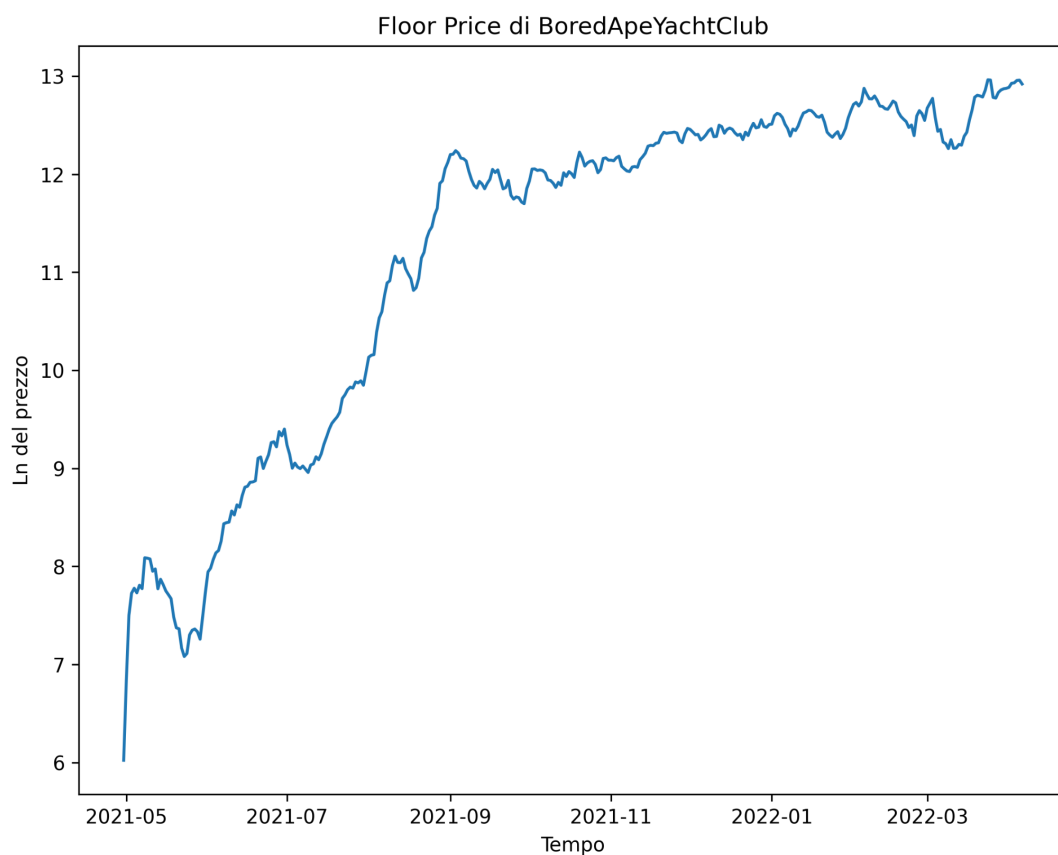


Tale funzione permette di **linearizzare** l'andamento del grafico, abbassando notevolmente l'ordine di grandezza del suo argomento (le cifre grosse diventano piccole, ad esempio  $\log_e(1000) = 6.90$ ,  $\log_e(100000) = 11.51$  ecc..)

Ecco il **codice** dove l'andamento dei dati viene linearizzato:

```
data['floor_price_quote_7d'] = np.log(data['floor_price_quote_7d'])
```

Qui possiamo notare come diminuendo la varianza **cambia** il grafico:



Abbiamo deciso di utilizzare questo processo di linearizzazione dei dati per via dei migliori risultati ottenuti e perché è un processo **molto comune** negli studi predittivi su grafici temporali con grandi varianze, eseguito nella quasi totalità della documentazione presa in esame.

Da qui in avanti per i modelli verranno utilizzati i **dati con varianza ridotta**, così da avere risultati migliori.

## Metodi proposti

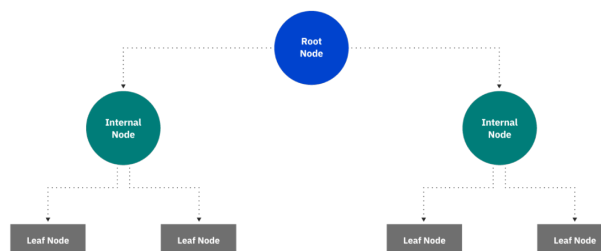
Per arrivare alla **scelta** definitiva di utilizzare il modello ARIMA, abbiamo valutato diversi modelli, tra cui:

### 1. Alberi di regressione

Come prima cosa abbiamo deciso di provare ad utilizzare un albero di regressione, cioè un **algoritmo di apprendimento supervisionato** non parametrico. Si compone di una struttura ad albero gerarchica così composta:

- ogni **nodo interno** rappresenta una variabile;
- un **arco verso un nodo figlio** rappresenta un possibile valore per quella proprietà;
- una **foglia** rappresenta il valore predetto per la classe a partire dai valori delle altre proprietà, che nell'albero è rappresentato dal cammino (path) dalla nodo radice (root) al nodo foglia.

Un albero di regressione viene costruito utilizzando tecniche di apprendimento a partire dall'insieme dei dati iniziali (training set) per i quali è nota la classe.



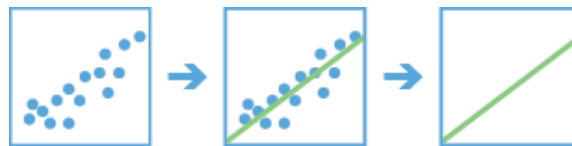
Gli alberi di regressione funzionano meglio con una **larga quantità di attributi** e uno dei più grandi **svantaggi** di questo modello è che la **complessità spaziale** dell'algoritmo potrebbe diventare **proibitiva** e avendo a che fare con problemi complessi lo spazio delle ipotesi diventa troppo grande. Per questo motivo abbiamo deciso di utilizzare **altri modelli** e non siamo passati all'implementazione di questo modello nel nostro programma.

## 2. Modello AutoRegressivo

Il modello autoregressivo viene indicato con **AR(p)** dove p è l'ordine del modello, è la rappresentazione di un tipo di processo stocastico; come tale descrive alcuni processi che variano nel tempo come l'economia, ecc.

Questo modello si usa per stimare un valore **numerico reale** sulla base di altri attributi: ad esempio, devo assegnare il valore a un'auto usata sulla base di un database di prezzi a cui ho venduto in precedenza auto usate.

Il modello autoregressivo è un modello **lineare** che specifica che la variabile in uscita dipende linearmente dai valori delle uscite precedenti.



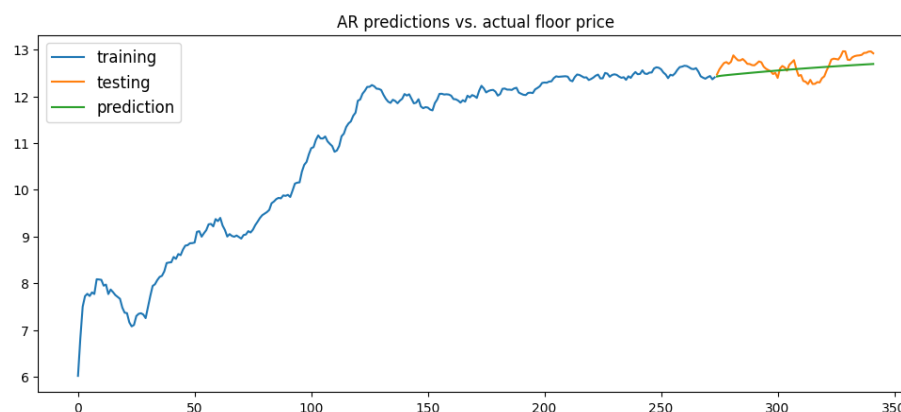
La **regressione** risolve il problema di una relazione funzionale tra variabili misurate sulla base di dati campionari estratti da un'ipotetica popolazione infinita.

La libreria che abbiamo utilizzato per includere questo modello al nostro progetto è stata [statsmodels.tsa.ar\\_model](#).

Il codice che richiama questo modello è il seguente:

```
a_res = AutoReg(a_train, lags=ar_select_order(a_train, maxlag=29, ic='aic').ar_lags).fit()
a_res.predict(start=datetime(2022, 1, 29), end=datetime(2022, 4, 6), dynamic=False)
```

Utilizzando il modello AR attraverso il programma scritto in Python abbiamo fatto questa previsione sulla collezione di Bored Ape Yacht Club.



Abbiamo notato che tramite questo modello riuscivamo a capire la **tendenza** (ribassista o rialzista) di una collezione di NFT e avevamo come risultato un valore che si **avvicinava** di molto **al prezzo reale**, con un margine di errore solo dell'1,5%. Attraverso lo studio di questo modello ci siamo accorti dell'esistenza di modelli più avanzati e sperando di ottenere un risultato migliore non ci siamo fermati qui.

### 3. Modello ARMA

Per verificare al meglio se esistevano modi migliori per predire il Floor Price di una collezione di NFT, abbiamo provato a considerare un modello più avanzato rispetto al modello AR, cioè il modello **ARMA** (*AutoRegressive Moving Average*) che **estende** il modello autoregressivo considerando gli errori come serialmente correlati. Formalmente si dice che una serie storica  $x_t$  segue un modello ARMA( $p, q$ ) se soddisfa la relazione:

$$X_t = \alpha + \overset{\text{AR}}{\varrho_1 X_{t-1} + \dots + \varrho_p X_{t-p}} + \varepsilon_t + \overset{\text{MA}}{\varphi_1 \varepsilon_{t-1} + \dots + \varphi_p \varepsilon_{t-q}}$$

Il modello ARMA può essere considerato come un modo per **approssimare** le autocovarianze di  $x_t$ , perché qualunque serie temporale  $x_t$  con covarianza finita può essere scritta come un AR o come un MA con errori non correlati.

Come per il modello autoregressivo, la scelta dell'ordine del modello deve rispondere alle esigenze contrapposte di un buon adattamento ai dati e di parsimonia nel numero di parametri da stimare.

Nel caso in cui i dati evidenzino la presenza di non stazionarietà, è talvolta possibile rimuovere tale non stazionarietà attraverso la trasformazione in differenze prime,  $x_t - x_{t-1}$ .

Il modello ARMA( $p, q$ ) applicato ai dati così trasformati prende il nome di modello **ARIMA** (Autoregressive Integrated Moving Average) con parametri ( $p, 1, q$ ). La trasformazione dei dati in differenze prime può essere applicata  $d \geq 0$  volte, ottenendo così il modello ARIMA( $p, d, q$ ). In particolare, il modello ARIMA( $p, 0, q$ ) coincide con il modello ARMA( $p, q$ ).

Arrivati a questo punto abbiamo deciso di utilizzare il modello ARIMA, così da riuscire ad avere un risultato più preciso.

## 4. Modello ARIMA

Il modello ARIMA è la classe più generale dei modelli per la previsione di una serie temporale che può essere resa "stazionaria".

Il modello ARIMA(p, d, q) ha tre componenti:

1. **p** è il numero di termini auto-regressivi
2. **d** è il numero di differenze necessarie per la stazionarietà
3. **q** è il numero di errori nell'equazione di previsione

Per decidere quali valori di p, d e q utilizzare nel modello ARIMA, è utile dare un'occhiata ai dati sui **prezzi minimi** stessi, alla sua prima differenza (alla sua seconda differenza o superiori) insieme alla sua funzione di autocorrelazione ACF e alla funzione di autocorrelazione parziale PACF.

La previsione del prezzo dei token può anche essere eseguita acquisendo dati a livello di token, come tratti, transazioni a livello di token e **prezzi**.

Poiché il prezzo di vendita passato è di solito un buon indicatore del prezzo futuro, questo può essere interpretato come la previsione di un evento dipendente dal **tempo**:

$$Y(c|t_n) = f(w, x, Y(c|t_0, \dots, t_{n-1}))$$

- Y è il prezzo minimo della raccolta c al tempo  $t_n$
- x rappresenta le variabili dipendenti dal tempo
- Y da  $t_0$  a  $t_{n-1}$  rappresenta i prezzi passati che possono essere utilizzati per modellare il prezzo ora

Sappiamo che i prezzi di molti NFT sono saliti alle stelle l'anno scorso e c'è sicuramente una tendenza **rialzista** nel prezzo. Dato che la maggior parte dei modelli delle serie temporali richiede stazionarietà (nessuna tendenza) nei dati, generalmente Y deve essere trasformato in una differenza per rimuovere l'andamento del prezzo:

$$dY(c|t) = f(w, x, dY(c|t_0, \dots, t_{n-1}))$$

dove dY è la variazione percentuale del prezzo della raccolta (o token) c dal mese t - 1 a t.

Per testare al meglio le prestazioni del modello ARIMA, l'intero set di dati è **suddiviso** in 80% per l'allenamento e il 20% per i test.

Poiché ci sono tre diversi parametri p, d, q nel modello ARIMA, le combinazioni delle scelte dei parametri potrebbero diventare molto grandi. Fortunatamente la libreria che abbiamo utilizzato ([pmdarima](#)) fornisce un ARIMA **automatizzato** che utilizza un approccio graduale per cercare più combinazioni di parametri p, d, q e scegliere il modello migliore.

```
Performing stepwise search to minimize aic
ARIMA(2,2,2)(0,0,0)[0] intercept : AIC=-523.985, Time=0.33 sec
ARIMA(0,2,0)(0,0,0)[0] intercept : AIC=-472.435, Time=0.02 sec
ARIMA(1,2,0)(0,0,0)[0] intercept : AIC=-510.965, Time=0.03 sec
ARIMA(0,2,1)(0,0,0)[0] intercept : AIC=-525.557, Time=0.05 sec
ARIMA(0,2,0)(0,0,0)[0] : AIC=-474.224, Time=0.01 sec
ARIMA(1,2,1)(0,0,0)[0] intercept : AIC=-526.478, Time=0.09 sec
ARIMA(2,2,1)(0,0,0)[0] intercept : AIC=inf, Time=0.27 sec
ARIMA(1,2,2)(0,0,0)[0] intercept : AIC=-524.562, Time=0.17 sec
ARIMA(0,2,2)(0,0,0)[0] intercept : AIC=-525.953, Time=0.17 sec
ARIMA(2,2,0)(0,0,0)[0] intercept : AIC=-514.758, Time=0.08 sec
ARIMA(1,2,1)(0,0,0)[0] : AIC=-527.752, Time=0.09 sec
ARIMA(0,2,1)(0,0,0)[0] : AIC=-526.861, Time=0.04 sec
ARIMA(1,2,0)(0,0,0)[0] : AIC=-512.508, Time=0.01 sec
ARIMA(2,2,1)(0,0,0)[0] : AIC=-529.378, Time=0.18 sec
ARIMA(2,2,0)(0,0,0)[0] : AIC=-516.221, Time=0.04 sec
ARIMA(3,2,1)(0,0,0)[0] : AIC=-526.396, Time=0.11 sec
ARIMA(2,2,2)(0,0,0)[0] : AIC=-528.154, Time=0.20 sec
ARIMA(1,2,2)(0,0,0)[0] : AIC=-525.829, Time=0.10 sec
ARIMA(3,2,0)(0,0,0)[0] : AIC=-523.732, Time=0.05 sec
ARIMA(3,2,2)(0,0,0)[0] : AIC=-524.613, Time=0.21 sec

Best model: ARIMA(2,2,1)(0,0,0)[0]
Total fit time: 2.278 seconds
```

Nel nostro caso, il modello ARIMA **migliore** risultante da diversi calcoli dal pacchetto 'pmdarima' è stato **ARIMA(2,2,1)(0,0,0)[0]**.





















Questo è il **codice** in cui viene richiamato il modello ARIMA:

```
model = pm.auto_arima(train, start_p=1, start_q=1,
                      max_p=5, max_q=5,
                      # m=1,
                      d=None,
                      seasonal=False,
                      trace=True,
                      suppress_warnings=True,
                      stepwise=True,
                      random_state=42)
```

Osservando i risultati avuti tramite questo modello, abbiamo compreso che nel nostro caso era la scelta **migliore** tra quelle viste in precedenza.

## Risultati sperimentali

I dati che abbiamo utilizzato per il nostro programma si basano sui dati storici ottenuti tramite le API di **Covalent** dalle diverse blockchain. È possibile scegliere una qualsiasi collezione presente in una tra le 40 blockchain disponibili su Covalent e utilizzarne i dati.

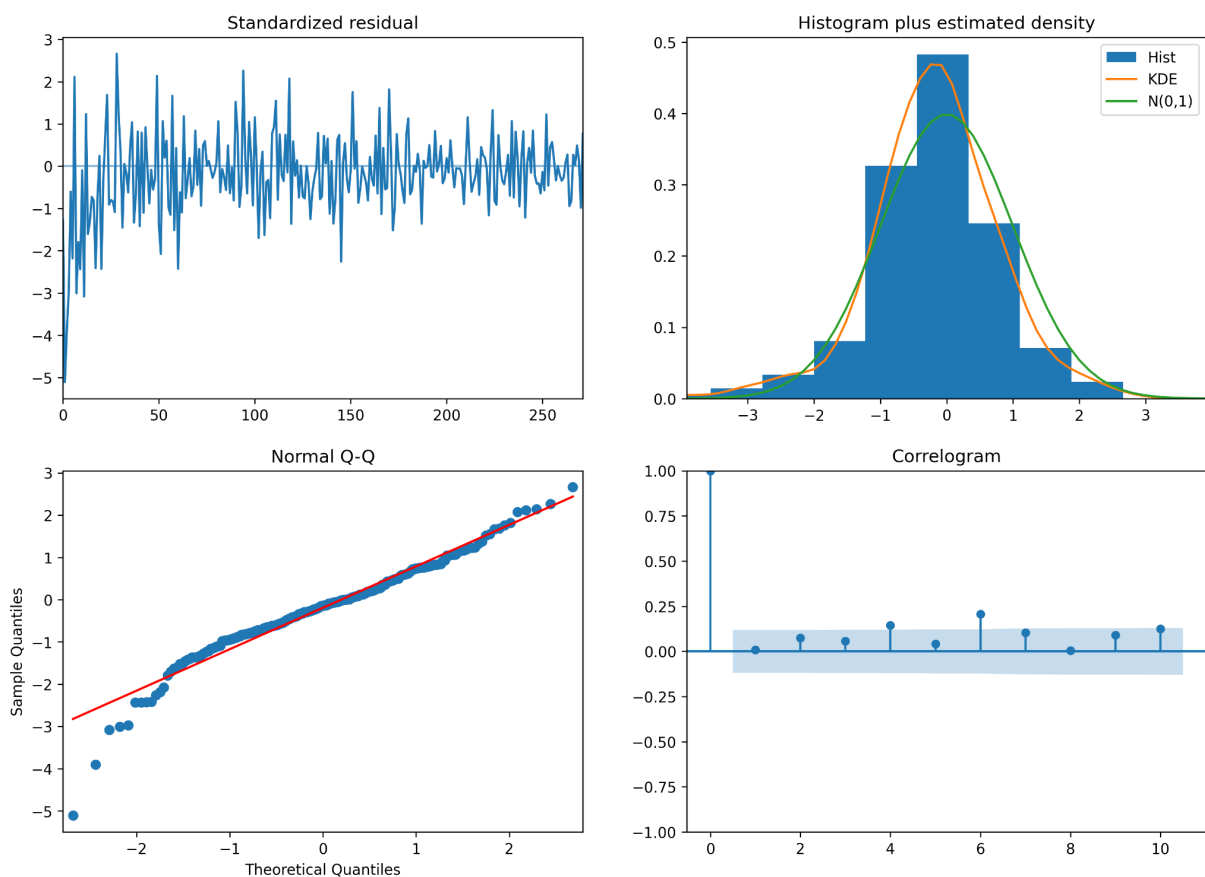
Blockchains Supported					
 ethereum	 polygon	 AVALANCHE	 BINANCE SMART CHAIN	 fantom	 Ronin
 ARBITRUM	 Moonbeam	 Moonriver	 palm	 Klaytn	 IoTeX
 HECO	 GODWOKEN	 rsk	 SHIDEN	 Evmos	 Harmony
 ASTAR	 cronos				

Il **software** che abbiamo utilizzato è stato scritto in **Python** ed è disponibile per eventuali test [qui](#). È possibile **testarlo** eseguendo in sequenza (dall'alto verso il basso) il codice presente nel notebook. Il software si trova attualmente alla quinta versione, perchè sono cambiati diversi modelli e metodi utilizzati.

Le **performance** sono state **valutate** attraverso lo studio del Mean Absolute Percentage Error: minore è il numero riportato da questo parametro, tanto maggiore sarà l'accuratezza

Una volta eseguito il modello, controlliamo anche i **dati residui** per verificare se tutte le ipotesi dei modelli ARIMA sono soddisfatte. Vengono controllati quattro componenti dei dati residui:

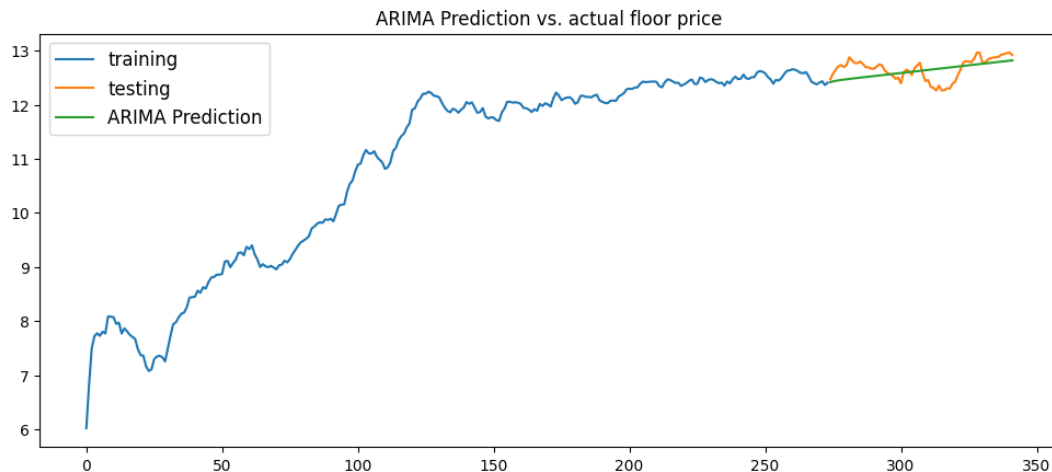
1. **Residuo standardizzato:** gli errori fluttuano intorno a una media di zero e hanno una varianza uniforme costante.
2. **Densità:** la densità empirica e la stima della densità kernale (KDE) delle serie di prezzi del piano suggeriscono che ha una media pari a zero ma ha una forma più sottile e appuntita rispetto a una distribuzione normale.
3. **Grafico QQ:** il grafico del prezzo minimo rispetto a una distribuzione normale mostra che non tutti i punti sono intorno alla linea rossa. Le deviazioni verso i valori negativi implicano che la distribuzione è distorta, la stessa conclusione dal grafico a densità.
4. **Correlogramma:** il grafico (ACF) mostra che la maggior parte dei punti rientra nell'intervallo di confidenza, quindi gli errori residui non sono autocorrelati. Tuttavia, c'è un punto che spinge lontano sul lato negativo, il che indica che potrebbe esserci qualche modello nei residui che non sono spiegati nel modello. L'aggiunta di più predittori (variabili esplicative) potrebbe aiutare a migliorare il modello.





## Risultati della configurazione migliore

Utilizzando la **configurazione migliore**, cioè il modello ottimale ARIMA(2,2,1), addestrato dai dati storici dal 30 aprile 2021 al 28 gennaio 2022, è possibile prevedere il prezzo minimo dal 29 gennaio 2022 al 6 aprile 2022.

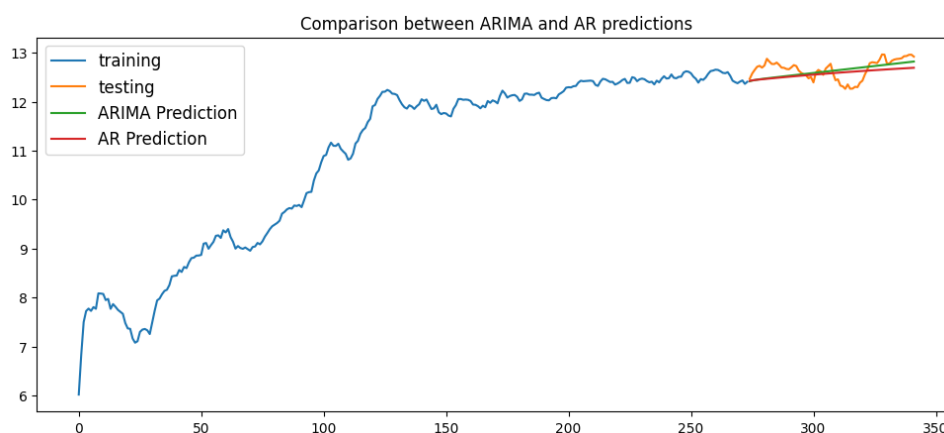


Le metriche delle prestazioni del modello confrontano il prezzo minimo previsto e effettivo durante il periodo di test dal 29 gennaio 2022 al 6 aprile 2022. L'**errore** percentuale assoluto medio è del 1,34%, significa che il modello è accurato di circa il 98,76% nel prevedere i prossimi 67 giorni.

```
{'Mean absolute percentage error': 0.013477515462762948,  
'Root mean squared error': 0.20478921001229305,  
'corr': 0.23385076909062547}
```

## Studio di ablazione

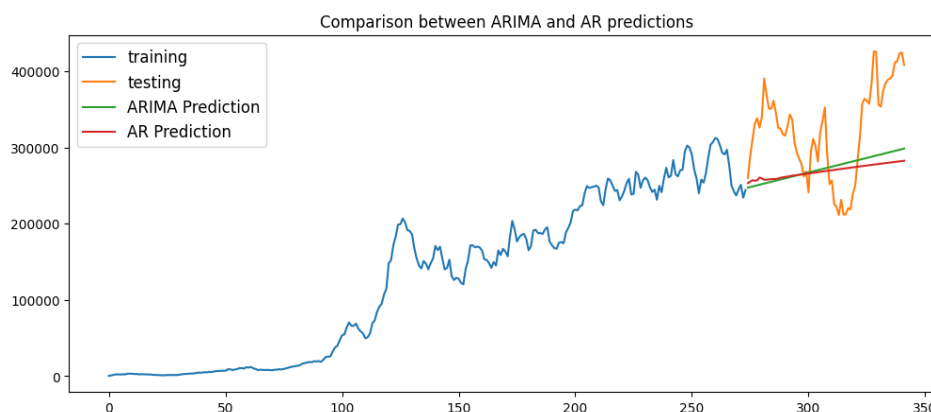
Mettendo a confronto il modello AR con il modello ARIMA (i due modelli più efficaci) e utilizzando i prezzi convertiti attraverso la funzione logaritmica, possiamo osservare che il **modello ARIMA**, anche se di poco (si tratta dello 0,16% di errore in meno), si avvicina di più al prezzo che è stato poi raggiunto dalla collezione di Bored Ape Yacht Club e quindi possiamo dire che è il più affidabile.



Questo è l'errore di AR, cioè l'1,5%, poco più alto rispetto all'1,34% del modello ARIMA:

```
{'Mean absolute percentage error': 0.015082792750851297,  
'Root mean squared error': 0.21458052689828594,  
'corr': 0.18761886429290056}
```

Possiamo notare dal grafico seguente come cambia la predizione (rispetto al grafico precedente) se non passiamo lo storico dei prezzi attraverso la funzione logaritmica: qui abbiamo un margine di errore **molto più elevato** sia per quanto riguarda il modello ARIMA, che cambia del 19,89% rispetto all'1,34% che abbiamo dopo aver convertito i prezzi attraverso la funzione logaritmica, sia per il modello AR che passa dal 20,10% all'1,5%.



Questo è l'errore di ARIMA(0,1,1), la versione ottimale per quanto riguarda i dati in entrata in formato WEI (corrisponde ad un altro metodo di predizione chiamato “exponential smoothing”, che utilizza le medie storiche di una variabile in un periodo per cercare di prevedere il suo comportamento futuro), cioè il 19,89%:

```
{'Mean absolute percentage error': 0.19894706076688498,  
'Root mean squared error': 74919.89346455934,  
'corr': 0.2957063761541203}
```

Questo invece è l'errore di AR, cioè il 20,10%:

```
{'Mean absolute percentage error': 0.2010614276945184,  
'Root mean squared error': 78154.87649467416,  
'corr': 0.28468450753612023}
```

Come vediamo in entrambi i casi il modello AR si dimostra di poco **inferiore** nella precisione rispetto al modello ARIMA.

Il fatto che spicca di più è la **differenza** nella precisione di entrambi i modelli attraverso l'utilizzo di dati in formati diversi:

1. Nel caso d'uso della funzione di logaritmo possiamo vedere una **precisione molto elevata**, grazie al livellamento della varianza che viene effettuato.
2. Nell'utilizzo dei dati con l'unità di misura WEI, la precisione si abbassa di molto, per via dell'**elevata varianza** che troviamo nei prezzi.

## Studio di comparazione

Nel documento di [Ho et al.](#) il loro modello (XGBoost), che prende in input tanti attributi degli NFT che considera, ha una precisione massima del 84.9% quando considera tutti gli attributi di utilità, quattro feature di rarità e quattro feature di base. Il nostro modello ARIMA (2,2,1) ha un'accuratezza ben **superiore** al 98,76%.

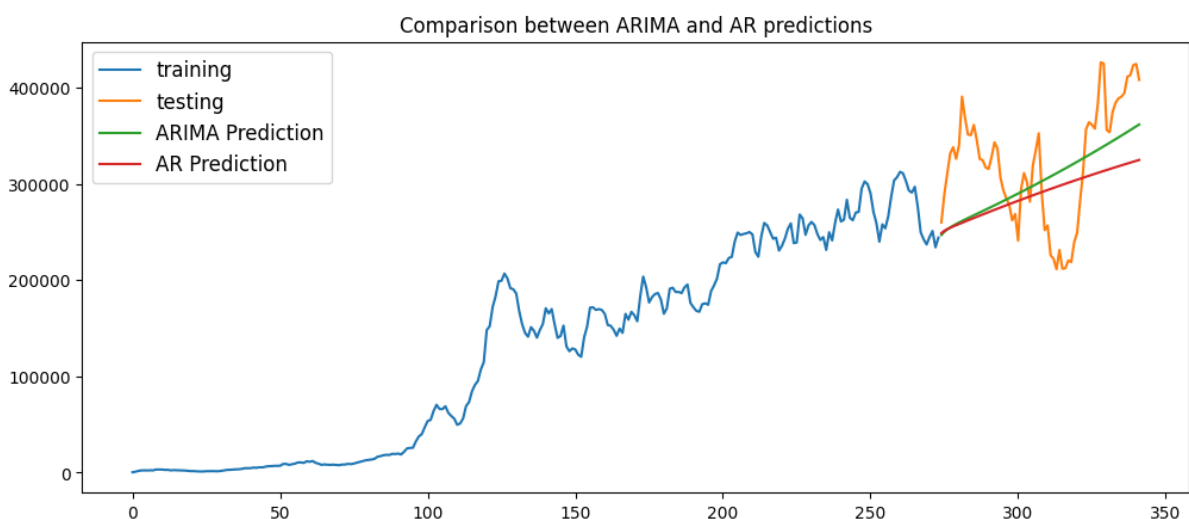
## Conclusione

Dai risultati che abbiamo ottenuto capiamo che **è possibile prevedere** con un certa sicurezza il valore futuro di una collezione di NFT.

D'altronde bisogna considerare che il futuro rimane imprevedibile e il prezzo degli NFT è influenzato anche dalle idee che hanno le persone e sulla loro psicologia, che cambia molto spesso con il tempo, quindi consideriamo sempre un margine d'errore e i cosiddetti “**cigni neri**” (eventi non previsti, che hanno effetti rilevanti e che, a posteriori, vengono inappropriatamente razionalizzati e giudicati prevedibili con il senno di poi) che possono presentarsi.

Il metodo proposto **rispetta** l'attesa di prevedere il prezzo futuro di una collezione di NFT, con una sorprendente precisione del 98,76%. Questi risultati ottenuti hanno rispecchiato la teoria dietro al modello scelto anche se non ci si aspettava dati così performanti. Il grande miglioramento si è infatti ottenuto utilizzando i dati sulla scala logaritmica.

Infine abbiamo voluto trasformare di nuovo i dati dei prezzi nel loro valore in WEI per verificare come appariva il grafico sulla previsione e questo è il risultato:



## Limiti del metodo

I limiti che abbiamo notato sono i seguenti:

1. Un **numero relativamente basso** di dati viene utilizzato per prevedere la collezione di NFT di BAYC mostrata negli esempi: probabilmente non ci sono stati abbastanza cicli (alti e bassi) nella storia del floor price per allenare il modello a prevedere in modo migliore il futuro.
2. ARIMA è un modello **generico** e semplice di serie temporali. Potrebbero esserci altri modelli e anche altre variabili esogene che possono essere utilizzate per costruire un modello migliore e avere risultati migliori.

## Lavori futuri

In futuro è possibile utilizzare il modello **SARIMAX** (*Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors*) che è una versione aggiornata del modello ARIMA: possiamo dire che SARIMAX è un modello stagionale equivalente come SARIMA e ARIMA. La caratteristica del modello che differisce dagli altri modelli è che può anche utilizzare **effetti esterni**, come ad esempio il ciclo economico in cui ci troviamo, ecc.

## Sitografia



- [NFT-Price-Prediction-CNN - Using visual feature extraction, prices of NFTs are predicted via CNN \(Alexnet and Resnet\) architectures. | PythonRepo](#)
- <https://anyfty.medium.com/anyftys-first-results-in-nft-pricing-prediction-68c131501524>
- [Mapping the NFT revolution: market trends, trade networks and visual features, Nadini et al.](#)
- [Covalent Unleashes New Real-Time APIs to Drill Down into NFT Markets - Covalent \(covalenthq.com\)](#)
- [Come riconoscere un Nft di valore - la Repubblica](#)
- [Cos'è un Albero Decisionale | IBM](#)
- [OpenSea, the largest NFT marketplace](#)
- [Regressione lineare - Wikipedia](#)
- [Modello autoregressivo - Wikipedia](#)
- [statsmodels.tsa.ar\\_model.AutoReg — statsmodels](#)
- [Exponential smoothing - Wikipedia](#)
- [Autoregressive integrated moving average - Wikipedia](#)
- [pmdarima · PyPI](#)
- [Complete Guide To SARIMAX in Python for Time Series Modeling \(analyticsindiamag.com\)](#)
- [NFT Appraisal Prediction: Utilizing Search Trends, Public Market Data, Linear Regression and Recurrent Neural Networks](#)
- [Durban Rodrigues, Marc. \*Development of a dataFrame and a Bot to predict NFT-collection performance\*. BS thesis. Universitat Politècnica de Catalunya, 2022.](#)
- [Modello ARMA - Cos'è, definizione e concetto - 2021 - Economy-Wiki.com](#)
- [Ho, Kin-Hon, et al. "Analysis of Non-Fungible Token Pricing Factors with Machine Learning."](#)